

UNIVERSIDAD AUTONOMA DE MADRID

ESCUELA POLITECNICA SUPERIOR



Grado en Ingeniería Informática

TRABAJO FIN DE GRADO

ENCODE: Tests online para el aprendizaje de C

Alfonso Bonilla Trueba
Tutor: Alejandro Sierra Urrecho

Abril 2020

ENCODE: Tests online para el aprendizaje de C

AUTOR: Alfonso Bonilla Trueba
TUTOR: Alejandro Sierra Urrecho

Grupo de la EPS Ingeniería Informática

Escuela Politécnica Superior
Universidad Autónoma de Madrid
Febrero de 2020

Agradecimientos

No sé por dónde empezar, porque todo lo que voy a mencionar ha sido muy importante en esta etapa.

Primero quiero empezar por dar las gracias a mis padres por todo el esfuerzo que han hecho para que yo pueda estar hoy aquí finalizando mis estudios, tanto a nivel económico como personal por siempre confiar en mí. Desde pequeño siempre quise estudiar informática y aquí estoy hoy.

A mis amigos, en especial a Mónica, Javier y Jorge, ellos han sido más que amigos en esta etapa, me han aguantado mucho y yo también a ellos. Otro gran apoyo ha sido Alexander, ha tenido una paciencia increíble y ha confiado siempre en mí y en la etapa final de este proyecto ha sido mi apoyo número uno.

Agradecer también a la Universidad Autónoma de Madrid todos los recursos, tanto físicos como pueden ser los laboratorios, aulas, biblioteca, como humanos, profesores que se implican con los alumnos. También agradecer a Alejandro Sierra, tutor de este trabajo, por la gran ayuda aportada e ideas. Todo esto en conjunto ha hecho posible llegar hasta aquí, llegar a ser Ingeniero Informático.

Resumen (castellano)

El objetivo de este Trabajo Fin de Grado es ampliar la aplicación web de aprendizaje de lenguajes de programación, dotándola de un módulo para examinar e ir adaptando las pruebas en función de las necesidades de cada alumno. Un objetivo secundario de este trabajo ha sido mejorar la usabilidad, con colores más amigables e interfaz simplificada.

Esta herramienta al igual que su primera versión podrá ser utilizada en cualquier asignatura de la Escuela Politécnica Superior de la UAM. Con esta nueva ampliación se abre la puerta a que pueda ser utilizada en otro tipo de asignaturas y no solo exclusivamente en aquellas que tienen contenido de programación como ocurre en la actualidad con cursos de C, Kotlin y Python.

Este nuevo módulo continúa con un aspecto clave de la aplicación base, ha sido diseñado y programado para que pueda ser ampliado fácilmente y que tenga un mantenimiento sencillo. La filosofía seguida para su desarrollo ha sido no centrarse en la nota obtenida en los exámenes, sino en la mejora del aprendizaje del estudiante. En lugar de utilizar preguntas de tipo test para evaluar al estudiante, se utilizan para conocer su evolución y reforzar su aprendizaje.

Además del diseño y programación del nuevo módulo para la herramienta, se han mejorado varios puntos del resto de la plataforma. Mejora de la interfaz, haciéndola más amigable y serena para los estudiantes, mejora de la usabilidad al cambiar los menús de navegación. Por último, pero no menos importante, se ha mejorado la seguridad, evitando los accesos por parte de los alumnos a los cuadros de control para el administrador y corrigiendo errores

En este documento se explicarán tanto los aspectos de diseño, como los aspectos técnicos empleados en el desarrollo, así como todos los conocimientos adquiridos durante la carrera que han hecho posible la realización de este proyecto.

Palabras clave (castellano)

Django, Framework, Aplicación web

Abstract (English)

The objective of this Final Degree Project is to expand the web application for learning programming languages, providing it with a module to examine and adapt the tests according to the needs of each student. A secondary objective of this work has been to improve usability, with more friendly colors and simplified interface.

This tool, like its first version, can be used in any subject of the UAM Polytechnic School. With this new extension, the door is opened for it to be used in other types of subjects and not only exclusively for those that have programming content as is currently the case with C, Kotlin and Python courses.

This new module continues with a key aspect of the base application, has been designed and programmed so that it can be easily expanded and easy to maintain. The philosophy followed for its development has been not to focus on the mark obtained in the exams, but on the improvement of student learning. Instead of using test questions to evaluate the student, they are used to know their evolution and reinforce their learning.

In addition to the design and programming of the new module for the tool, several points of the rest of the platform have been improved. Improved interface, making it more friendly and serene for students, improved usability when changing navigation menus. Last but not least, security has been improved, preventing access by the students to the control panels for the administrator and correcting errors

This document will explain both the design aspects, as well as the technical aspects used in the development, as well as all the knowledge acquired during the race that made this project possible

Keywords (English)

Django, Framework, Web Application

INDICE DE CONTENIDOS

1 INTRODUCCIÓN.....	1
1.1 MOTIVACIÓN	1
1.2 OBJETIVOS.....	1
1.3 ORGANIZACIÓN DE LA MEMORIA	2
2 ESTADO DEL ARTE	3
2.1 HERRAMIENTAS Y METODOLOGÍAS ACTUALES.....	3
2.2 TECNOLOGÍAS EMPLEADAS	3
2.2.1 Framework	3
2.2.2 Servidor web	4
2.2.3 Base de datos.....	4
2.2.4 Front-end.....	4
3 DISEÑO	5
3.1 MÓDULOS DE LA HERRAMIENTA.....	5
3.2 CÁLCULO DE PROBABILIDADES	5
3.3 ALGORITMO DE SELECCIÓN DE PREGUNTAS.....	6
3.4 BASE DE DATOS	7
3.4.1 Base de datos del módulo Editor.....	8
3.4.2 Base de datos módulo Tutorial	9
4 DESARROLLO.....	11
5 INTEGRACIÓN, PRUEBAS Y RESULTADOS	13
5.1 INTEGRACIÓN.....	13
5.2 PRUEBAS	13
5.2.1 Pruebas al cálculo de probabilidades de las categorías.....	13
5.2.2 Pruebas a la selección de preguntas	14
6 CONCLUSIONES Y TRABAJO FUTURO.....	17
6.1 CONCLUSIONES.....	17
6.2 TRABAJO FUTURO	18
BIBLIOGRAFÍA	19
GLOSARIO	21
ACRÓNIMOS	I
ANEXOS	II
A MANUAL DE USUARIO	II
A.1 MÓDULO TUTORIAL.....	II
A.2 MÓDULO EDITOR	VIII

INDICE DE FIGURAS

FIGURA 1: DIAGRAMA DE FLUJO PARA EL ALGORITMO DE SELECCIÓN.....	7
FIGURA 2: ESQUEMA RELACIONAL DEL MODELO DEL MÓDULO EDITOR	9
FIGURA 3: ESQUEMA RELACIONAL DEL MODELO DEL MÓDULO TUTORIAL.....	10
FIGURA 4: DOMINIOS OBTENIDOS POR LA PLATAFORMA.....	13
FIGURA 5: PRUEBA EXAMEN ALEATORIO.....	15
FIGURA 6: PRUEBA EXAMEN CON CONOCIMIENTOS	15
FIGURA 7: LOGIN ENCODE	II
FIGURA 8: REGISTRO ENCODE	II
FIGURA 9: PÁGINA PRINCIPAL DE ENCODE.....	III
FIGURA 10: CONFIRMACIÓN ABANDONO DE CURSO	III
FIGURA 11: LISTA DE UNIDADES Y EXAMENES.....	IV
FIGURA 12: INSTRUCCIONES DEL EXAMEN	IV
FIGURA 13: BOTÓN INSTRUCCIONES DEL EXAMEN.....	V
FIGURA 14: FORMATO EXAMEN.....	V
FIGURA 15: FEEDBACK EXAMEN.....	VI
FIGURA 16: REVISIÓN DE UN EXAMEN	VI
FIGURA 17: VOLVER AL CURSO DESPUÉS DE LA CORRECCIÓN	VII
FIGURA 18: MENSAJE REINTENTO PREMATURO DE EXAMEN.....	VII
FIGURA 19: MENÚ DESPLEGABLE PARA EL USUARIO	VIII
FIGURA 20: MENSAJE DE ACCESO RESTRINGIDO	VIII
FIGURA 21: VENTANA PARA EDICIÓN DE UN CURSO.....	IX
FIGURA 22: VENTANA PARA EDICIÓN, CREACIÓN Y ELIMINACIÓN DE CATEGORÍAS	IX
FIGURA 23: ERROR AL ELIMINAR UNA CATEGORÍA.....	X
FIGURA 24: CREACIÓN EXAMEN.....	X
FIGURA 25: EXAMEN RECIÉN CREADO.....	XI
FIGURA 26: CREACIÓN PREGUNTA.....	XI
FIGURA 27: PREGUNTA RECIÉN CREADA.....	XII
FIGURA 28: EDICIÓN DE UNA PREGUNTA.....	XIII

INDICE DE TABLAS

TABLA 1: PROBABILIDADES DE ACIERTO.....	6
TABLA 2: DESCRIPCIÓN DE LAS NUEVAS TABLAS Y ATRIBUTOS PARA EL MÓDULO EDITOR	9
TABLA 3: DESCRIPCIÓN DE LAS NUEVAS TABLAS Y ATRIBUTOS PARA EL MÓDULO TUTORIAL	9
TABLA 4: BOLSA DE PREGUNTAS EXAMEN DE PRUEBA	15
TABLA 5: ASIGNATURAS IMPRESCINDIBLES PARA EL DESARROLLO DEL PROYECTO	17

1 Introducción

Este TFG nace de la necesidad de ampliar la aplicación web **ENCODE** ya desarrollada previamente y puesta en funcionamiento para la Escuela Politécnica Superior. Dotarla de un módulo para poder evaluar a los alumnos y que estos vayan recibiendo *feedback* sobre su evolución.

El objetivo principal de este TFG es dotar de exámenes generados en función de las necesidades de cada alumno para una plataforma ya existente. A continuación, se irá exponiendo la base para la creación de este nuevo módulo.

1.1 Motivación

Actualmente las herramientas de aprendizaje online han evolucionado de manera que se adaptan a sus usuarios, han dejado de ser simples tutoriales con unas evaluaciones estáticas. Los exámenes pueden personalizarse para cada alumno, y generarse dinámicamente en función de las necesidades de cada alumno para reforzar sus conocimientos.

Se parte de una aplicación web ya en funcionamiento para el aprendizaje de lenguajes de programación. Esta herramienta dispone de unidades formadas por objetivos. En cada uno de estos objetivos hay pequeñas pruebas sobre el código que se está mostrando en dicho objetivo. Estas pruebas no sirven para evaluar correctamente, ni son personalizados según el alumno.

Otra gran motivación es, que ya que se está usando actualmente la plataforma para alguna asignatura del grado de ingeniería informática es altamente necesario mejorar la plataforma y dotarla de nuevas funcionalidades. Es por esto por lo que se decidió ampliar la plataforma para adecuarla a la competencia y que se mantenga usando en el ámbito de la Escuela Politécnica Superior y que siga creciendo hasta ser usada en otras asignaturas e incluso en otros grados.

Todos estos son los motivos por los que surgió la necesidad de ampliar la plataforma **ENCODE**.

1.2 Objetivos

Mantener la plataforma como una web innovadora y competitiva con las ya existentes, y que pueda ser usada por los profesores de la Escuela Politécnica Superior.

Es necesario marcar unos objetivos claros para poder abordar este trabajo. El primero de ellos es la ampliación que necesita la base de datos ya existente para poder soportar y almacenar todos los cambios necesarios. El segundo es que todas las modificaciones nuevas que se incluyan mantengan la seguridad con la que ya se desarrolló la aplicación base, y que la usabilidad sea simple e intuitiva, mejorando la interfaz existente y diseñando las nuevas funcionalidades con esta misma filosofía.

Se ha de continuar con la modularidad con la que se comenzó la plataforma, respetando el desarrollo separado entre la parte del alumno y la parte del editor (en este caso el profesor). Esto permitirá seguir ampliando la plataforma en el futuro e ir incluyendo nuevas

funcionalidades de manera ordenada y correcta que asegure siempre el mejor funcionamiento posible. Para poder generar la selección de las preguntas acorde a las necesidades de cada estudiante será necesaria la implementación de un algoritmo de selección basado en las probabilidades condicionadas de cada estudiante.

El objetivo final de este proyecto no es que los estudiantes se sientan que están siendo evaluados de la manera tradicional y que van obteniendo notas numéricas. El objetivo real es que sientan que los exámenes están ideados para que su aprendizaje sea adaptado y no se centren simplemente en aprobar.

En este documento se va a ir documentando y explicando cada una de las fases del desarrollo de este proyecto de ampliación de la plataforma **ENCODE**.

1.3 Organización de la memoria

El presente documento ha sido redactado siguiendo la normativa para los Trabajo de Fin de Grado y siguiendo la plantilla disponible. La memoria está compuesta por: introducción, estado del arte, diseño y desarrollo.

Para la introducción se comienza analizando las herramientas y metodologías empleadas en el desarrollo de la aplicación base. Así podemos asegurarnos de que la decisión tomada sigue siendo la correcta para una herramienta de este tipo.

En la sección de diseño se hablará sobre los frameworks empleados y de la modularidad tan necesaria que nos proporcionan. También se trata el diseño de la ampliación de la base de datos, manteniendo el funcionamiento de la parte ya existente e introduciendo las nuevas tablas y relaciones necesarias. En esta sección también se detalla el funcionamiento del algoritmo de selección de preguntas, que es el corazón de esta ampliación.

Después de mostrar y detallar el diseño del nuevo módulo para el proyecto, se comienza a detallar cada uno de los pasos de desarrollo seguidos, desde la primera toma de contacto con la herramienta ya en funcionamiento hasta la puesta en marcha del nuevo módulo. Todo este proceso sin interrumpir el normal funcionamiento de la plataforma.

Una vez detallado el proceso de desarrollo vienen las pruebas, una parte importante para asegurar que el desarrollo y el diseño planteados funcionan correctamente. Es vital asegurarse de que el algoritmo de selección de preguntas está programado y diseñado bien y la selección es correcta según las probabilidades. Otro punto que comprobar es que cuando se corrigen los exámenes el cálculo de las probabilidades es correcto, de estas dependerá después la selección de preguntas.

Como finalización de este documento se explicarán los pasos para poder disponer del entorno de desarrollo para futuras mejoras y ampliaciones de la plataforma. Además, también se incluye un manual de usuario para todas las nuevas funcionalidades incluidas.

2 Estado del arte

Antes de comenzar con las explicaciones técnicas del desarrollo de este trabajo es necesario abordar el tema de cómo surgió la idea. Se analizarán las metodologías y herramientas utilizadas de forma breve para explicar la ampliación de la plataforma de aprendizaje ya existente y las carencias detectadas.

2.1 Herramientas y metodologías actuales

Como ya se ha comentado anteriormente, en la actualidad las plataformas de aprendizaje online han evolucionado ofreciendo una metodología personalizada según las necesidades de sus usuarios, además de ofrecer exámenes para poder cuantificar el avance. La metodología usada no siempre es la misma y varía según la plataforma.

- MOOC (Masive Open Online Course) (e.g. Edx[1], Udemy[2])
- Aprendizaje autónomo de materias tecnológicas (e.g. CodeAcademy[3], W3Schools [4])
- Aprendizaje cooperativo (e.g. StackOverflow[5], Stack Exchange[6])

ENCODE está enfocado para el aprendizaje de lenguajes de programación. Los ejemplos anteriormente mostrados han avanzado mucho en los últimos años y lo siguen haciendo actualmente, pero ninguno de ellos está enfocado para ser usado internamente en una institución de enseñanza. El funcionamiento cooperativo de ENCODE lo hace diferente del resto de plataformas existentes, es por esto, que es necesario actualizarlo y dotarle de funcionalidades que ya llevan tiempo en el resto de las plataformas. El objetivo de este proyecto es añadir el módulo de exámenes adaptados a las necesidades de cada alumno.

2.2 Tecnologías empleadas

El proyecto como ya se ha comentado consiste en la ampliación de una plataforma ya existente desarrollada ya en otro proyecto, por lo tanto, ciertas decisiones de diseño y elección de tecnologías a usar no han tenido que ser tomadas. De haber tomado la decisión de cambiar la tecnología con la que está desarrollada ENCODE habría que haber vuelto a desarrollar todo desde el principio.

A continuación, vamos a detallar las tecnologías usadas y justificar la continuidad de seguir con su uso.

2.2.1 Framework

El framework web para la realización de este proyecto es Django[10], la decisión fue tomada cuando se realizó la primera versión de la que se parte para este proyecto. Su elección es la más adecuada puesto que su puesta en marcha en un servidor es muy sencilla y se ha estudiado en una asignatura durante la carrera. Además, Django es muy versátil puesto que funciona con distintos tipos de bases de datos y se puede complementar con otros para realizar la interfaz de la web.

La programación backend con Django se realiza con Python, lenguaje estudiado durante el Grado y que actualmente está en auge y con muchísimas librerías disponibles, lo que lo hacen una elección estupenda y versátil.

2.2.2 Servidor web

La aplicación de la que se parte está alojada en un servidor de la Universidad Autónoma de Madrid, **encode.eps.uam.es**. El desarrollo de este nuevo módulo no está alojado en ningún servidor, se encuentra de manera local, por tanto, no está accesible públicamente. En el apartado 5.1 se trata la integración del módulo nuevo a la herramienta existente en el servidor.

2.2.3 Base de datos

La gestión de datos está realizada mediante **PostgreSQL** continuando con la decisión tomada en el diseño inicial de la herramienta, puesto que este gestor ha sido empleado durante la carrera y es uno de los mejores gestores para Django [11].

2.2.4 Front-end

Durante la carrera, cada vez que hemos desarrollado la parte visual para una aplicación lo hemos hecho de manera básica sin usar ningún framework. En el desarrollo inicial se decidió el uso de Semantic-UI para el desarrollo del front-end, en esta segunda versión se ha continuado con esa decisión, pero se ha dado un ligero lavado de cara a la interfaz ya existente. Con estos ligeros cambios de interfaz se ha buscado conseguir que la aplicación sea más atractiva a los usuarios y sencilla de utilizar.

En el desarrollo del front-end de este nuevo módulo han sido utilizadas las siguientes tecnologías:

- **Framework Semantic-UI** [7]: Proporciona componentes básicos para la interfaz, como son menús desplegados, tablas, botones...
- **JQuery** [8]: Biblioteca de JavaScript que permite la modificación de contenido directamente en el cliente, por tanto, sin necesidad de recargar la página.
- **JQuery UI** [9]: Rama de JQuery que permite añadir efectos visuales.
- **Plugin Ace Editor** [12]: Herramienta que permite la creación y edición de código con una estética agradable y con un funcionamiento muy similar al de los editores de texto.

Aunque se podría haber migrado toda la interfaz a una tecnología más puntera, este cambio podría considerarse un proyecto en sí mismo. El hecho de continuar con el uso de estos frameworks asegura un fácil mantenimiento y expansión para el futuro, así como un funcionamiento y experiencia de usuario excelentes como se vio en la primera versión de la plataforma.

3 Diseño

Una vez explicadas las tecnologías empleadas en este proyecto se va a explicar la estructura del proyecto. Se van a mostrar las decisiones tomadas para su correcta ampliación tanto a nivel de funcionalidad e interfaz, como a nivel interno de base de datos.

3.1 Módulos de la herramienta

La herramienta sigue estando compuesta de dos módulos puesto que no ha habido cambios en el diseño de mantener por separado las funciones del administrador que permite crear y modificar el contenido de la web, y el módulo **tutorial** dedicado a los estudiantes que se matriculan en los cursos y realizan sus actividades.

A nivel backend, han sido ampliados ambos módulos con nueva funcionalidad. Para el módulo editor ha sido necesario incluir toda la funcionalidad para poder crear los exámenes, las preguntas, y las categorías necesarias para la segregación de los conocimientos asociados al curso y poder realizar las predicciones.

El módulo tutorial ha sido ampliado para poder mostrar estos nuevos exámenes, generarlos seleccionando las preguntas en función de los dominios del estudiante y poder corregirlos así como el cálculo de los dominios de las categorías ya mencionadas.

Los exámenes se plantean de la siguiente manera.

- Están compuestos por preguntas tipo test.
- Pueden contener código complementario al enunciado.
- Cada pregunta tiene 3 posibles respuestas.
- Solo se puede marcar una como correcta, aunque puede haber varias respuestas válidas.
- Cada unidad tiene asociado un máximo de un examen.
- No se proporciona una nota numérica al final de cada examen.
- Se muestra feedback a algunos usuarios con fines de investigación.
- Se pueden realizar tantas veces como se quiera, el algoritmo de selección garantiza que las preguntas serán diferentes cada vez.

Como el código complementario a cada pregunta es opcional, es decir, no es necesario introducirlo, asegura el principio de que la herramienta pueda ser usada en asignaturas sin contenido de programación.

3.2 Cálculo de probabilidades

Para poder realizar las predicciones y saber que categorías son las que necesita reforzar un alumno se hace uso de una red de Bayes. Para esto necesitamos tener ciertas probabilidades almacenadas e ir actualizándolas conforme el alumno avanza en el curso.

Para cada pregunta, se almacenan dos probabilidades, la probabilidad de acertar la pregunta condicionada por tener dominio sobre la categoría que trata, y la probabilidad de acertar condicionada por no tener dominio sobre la categoría. Estas probabilidades son estáticas y no van variando porque son específicas de la pregunta y no del alumno. Pueden

tomar 3 valores distintos que se dan en función de la dificultad seleccionada por el profesor cuando crea la pregunta.

	P (acertar dominio)	P (acertar no dominio)
Básica	90%	70%
Media	90%	60%
Avanzada	30%	20%

Tabla 1: Probabilidades de acierto

Adicionalmente, para cada alumno se almacena su dominio para cada categoría existente. Para evitar tener una tabla con un exceso de registros, el dominio solo se almacena para las categorías de preguntas que ya han sido contestadas.

Las probabilidades que se almacenan en esa tabla sí son dinámicas, es decir, con cada examen que realice el estudiante se actualizan siguiendo la siguiente fórmula en caso de acierto:

$$P(\text{Domino Actualizado}) = [P(\text{acertar} | \text{dominio}) * P(\text{dominio})] / Z$$

$$Z = [P(\text{acertar} | \text{dominio}) * P(\text{dominio})] + [P(\text{acertar} | \text{no dominio}) * P(\text{no dominio})]$$

En caso de fallo, la fórmula a seguir es muy similar:

$$P(\text{Domino Actualizado}) = 1 - [P(\text{acertar} | \text{no dominio}) * P(\text{no dominio})] / Z$$

$$Z = [(1 - P(\text{acertar} | \text{dominio})) * P(\text{dominio})] + [(1 - P(\text{acertar} | \text{no dominio})) * P(\text{no dominio})]$$

Este cálculo se realiza a medida que se va corrigiendo el examen y por tanto se van actualizando los dominios de las categorías examinadas, de tal manera que la próxima vez que se examine se seleccionarán las preguntas en función de estos dominios.

3.3 Algoritmo de selección de preguntas

La selección de las preguntas que conforman los exámenes se realiza de manera personalizada para cada alumno, por eso, se siguen los pasos del siguiente algoritmo (**Figura 1**):

- 1- Si no se dispone de conocimiento previo (Probabilidades de cada categoría) la selección se realiza de manera aleatoria tomando 10 preguntas del total disponibles para el examen que se va a realizar.
- 2- Si se dispone de conocimiento previo, se realiza la inversa proporcional para los dominios para las categorías que lleva peor el estudiante, estas son aquellas cuyo valor es inferior a 0.1. Se reparte entre estos el 80% de las preguntas, es decir, un máximo de 8 preguntas. Las preguntas que se seleccionan son aquellas que aún no han sido respondidas por el alumno.
- 3- El restante de preguntas que queden hasta completar el 100% se extraen de forma aleatoria tomando como bote para seleccionarlás las pertenecientes al examen actual o alguno de los anteriores, con esto se implementa la evaluación continua. La selección se toma con preguntas que el estudiante no haya contestado previamente.

- 4- Si con el paso 3 ya se ha completado el 100% del examen se da por acabada la selección de preguntas. Si aún falta alguna pregunta para poder tener el examen completo se toman aleatoriamente, aunque ya la haya contestado previamente en otro examen.

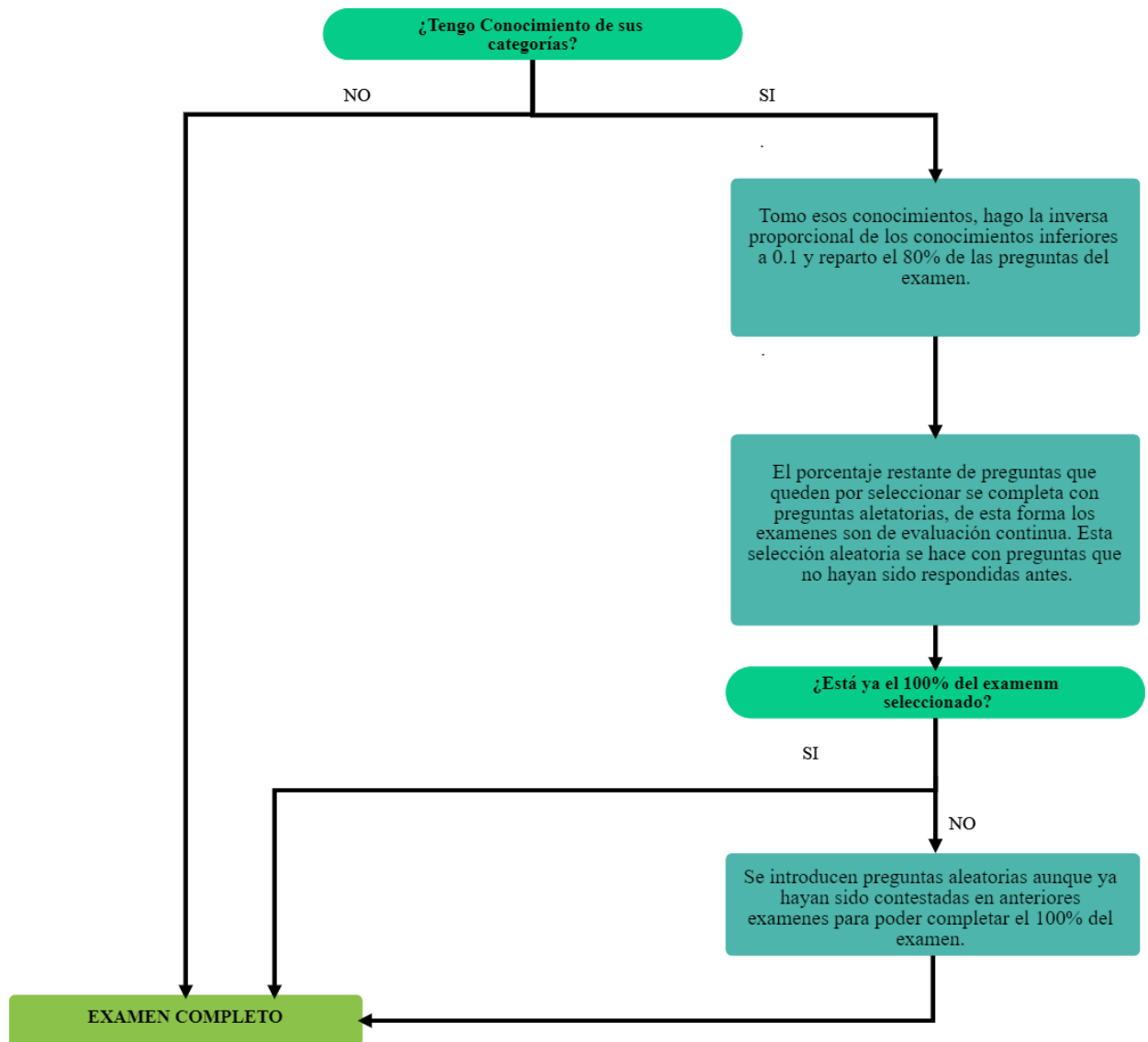


Figura 1: Diagrama de flujo para el algoritmo de selección

3.4 Base de Datos

Usando la abstracción proporcionada por Django se han creado las nuevas tablas necesarias, definidas mediante clases en Python. El diseño de esta ampliación no fue muy complicado, las tablas originales no han sido modificadas, pero si se han creado tablas nuevas para almacenar todos los datos necesarios para el nuevo módulo de exámenes, y estas nuevas tablas están directamente relacionadas con las ya existentes.

3.4.1 Base de datos del módulo Editor

Ha sido necesario ampliarla para poder almacenar todos los datos nuevos necesarios derivados del almacenamiento de los exámenes y preguntas, así como el almacenamiento de las respuestas dadas por los estudiantes. Estas tablas jerarquizan la información, dejando la nueva tabla de exámenes al mismo nivel que la de unidades ya existente.

En las siguientes tablas descriptivas de las nuevas tablas se omiten los campos generados automáticamente por Django, como puede ser el campo id.

Atributos	
UnitExam	<ul style="list-style-type: none">• <u>curso</u>: Curso al que pertenece el examen• <u>number_unit</u>: Número de unidad a la que pertenece el examen• <u>published</u>: booleano que indica si el examen esta publicado o no• <u>title</u>: Título del examen
ExamQuestion	<ul style="list-style-type: none">• <u>unitExam</u>: Examen al que pertenece la pregunta• <u>enunciate</u>: Enunciado de la pregunta• <u>p_correct_dom</u>: Probabilidad de acertar la pregunta si se tiene dominio sobre la categoría correspondiente• <u>p_correct_nodom</u>: Probabilidad de acertar la pregunta si no se tiene dominio sobre la categoría correspondiente• <u>tag</u>: Identificador de texto que indica el nivel y la categoría de la pregunta.
QuestionAnswer	<ul style="list-style-type: none">• <u>question</u>: Pregunta a la que pertenece la respuesta• <u>texto</u>: Respuesta• <u>isCorrectAnswer</u>: booleano que indica si la respuesta es correcta o no
QuestionCode	<ul style="list-style-type: none">• <u>question</u>: Pregunta a la que pertenece el código• <u>coding language</u>: Language de programación del código almacenado• <u>code</u>: Texto sobre el que se guarda el código
ExamScores	<ul style="list-style-type: none">• <u>user</u>: usuario (estudiante) al que pertenece la calificación• <u>exam</u>: Examen al que pertenece la calificación• <u>score</u>: Calificación obtenida• <u>passed</u>: booleano que indica si el examen ha sido superado o no• <u>time_stamp</u>: Fecha y hora en la que fue enviado el examen para corrección
TagDicc	<ul style="list-style-type: none">• <u>course</u>: Curso al que pertenece el diccionario• <u>description</u>: nombre de la categoría
LearnedTags	<ul style="list-style-type: none">• <u>user</u>: Usuario (estudiante) al que pertenece el dominio.• <u>course</u>: Curso al que está asociado el dominio

- description:
- p_domain: Dominio de la categoría

Tabla 2: Descripción de las nuevas tablas y atributos para el módulo Editor

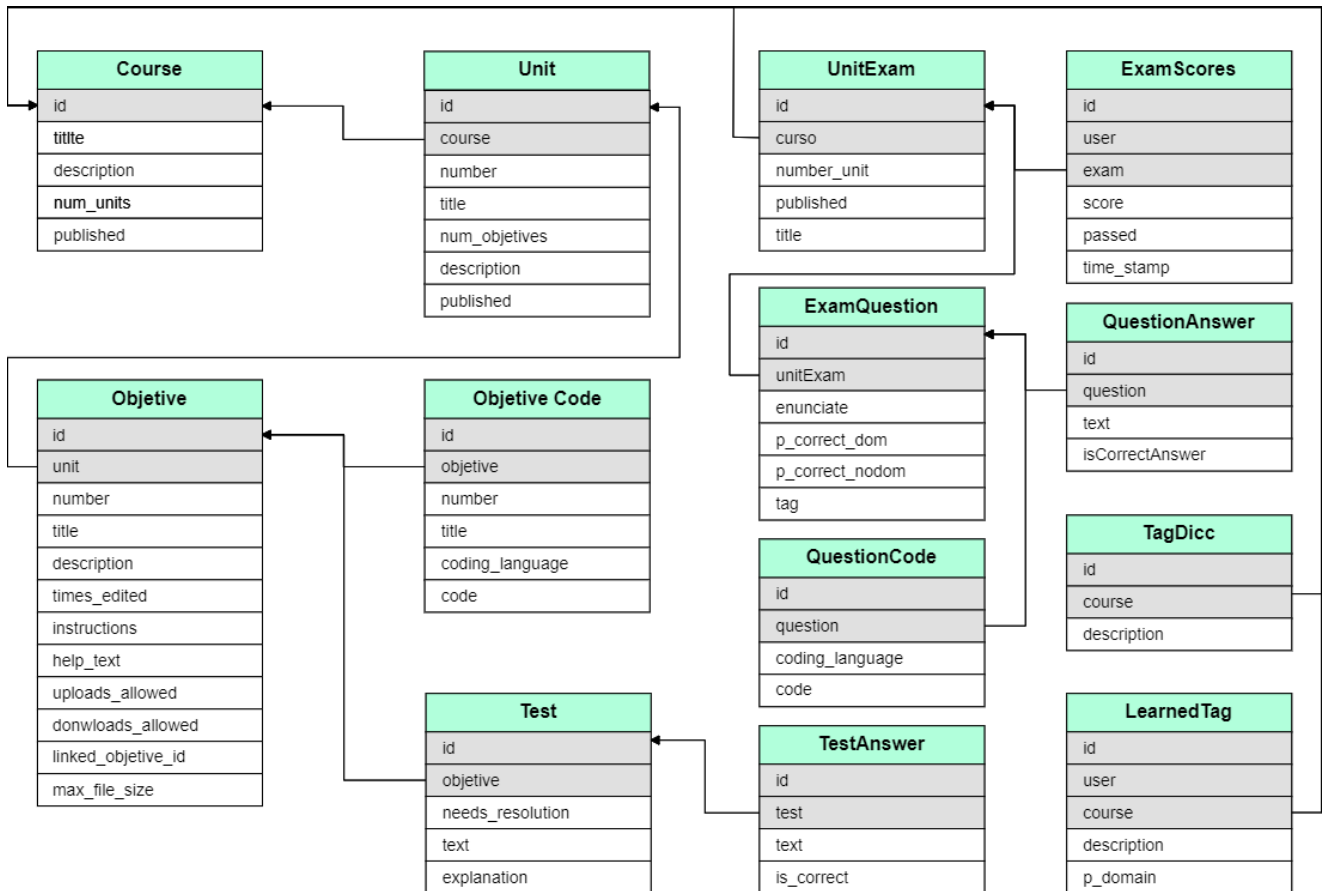


Figura 2: Esquema relacional del modelo del módulo Editor

3.4.2 Base de datos módulo Tutorial

La base de datos para el módulo Tutorial permanece prácticamente igual, puesto que dentro del objetivo de este trabajo no entraba ninguna modificación con la información que se maneja de los usuarios.

Se ha incluido una tabla para indicar a qué usuarios se les muestra feedback y a cuáles no. Cuando un usuario se da de alta de manera totalmente aleatoria se decide si se le va a proporcionar feedback al final de sus exámenes o no. Con el tiempo se irán acumulando datos para poder hacer un estudio de cómo afecta dar un feedback en lugar de una nota numérica en comparación con aquellos a los que no se les da.

Atributos	
StudentFeedback	<ul style="list-style-type: none"> • <u>user</u>: Uuario (estudiante) • <u>feedback</u>: booleano que indica si se muestra feedback después de los exámenes o no.

Tabla 3: Descripción de las nuevas tablas y atributos para el módulo Tutorial

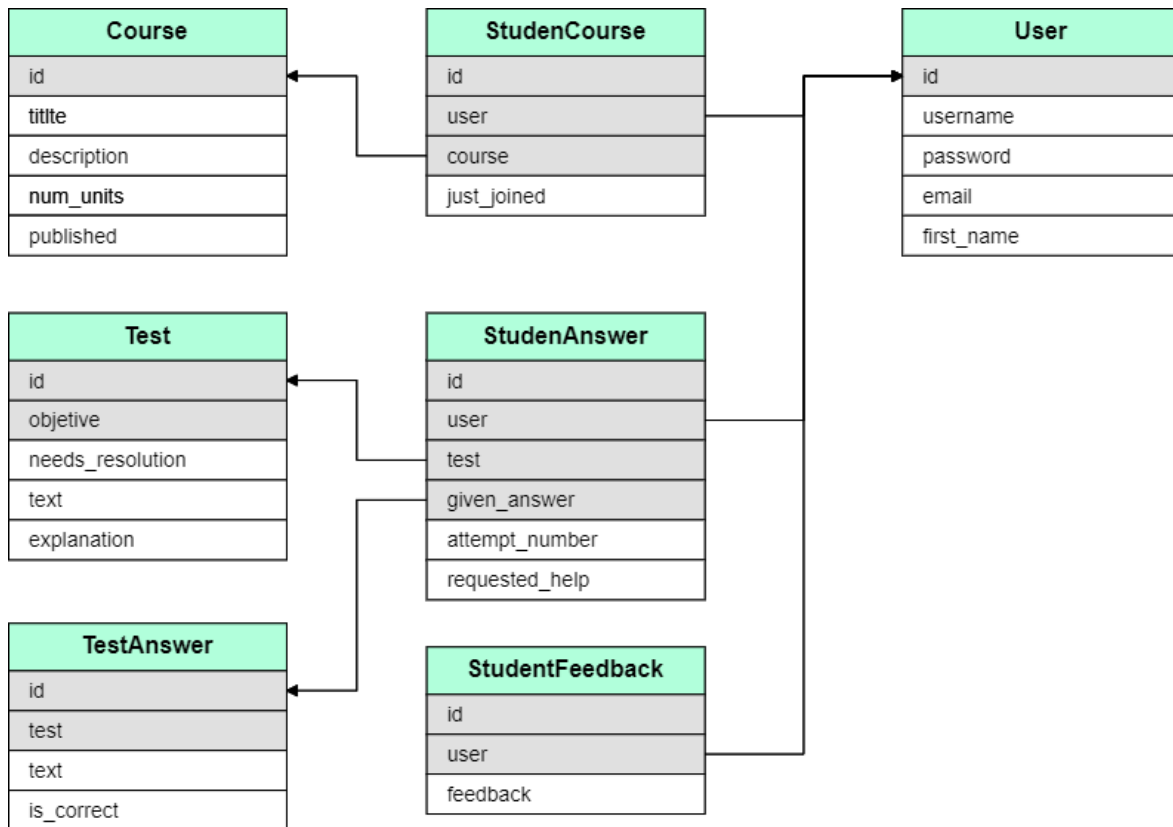


Figura 3: Esquema relacional del modelo del módulo Tutorial

4 Desarrollo

En este apartado se va a describir cómo fue el proceso de desarrollo sin entrar en detalles técnicos, puesto que todo el diseño ya ha sido detallado.

El desarrollo comenzó por arreglar todos los fallos detectados durante pruebas de funcionamiento y de cómo estaba diseñada la aplicación sobre la que se partía para el desarrollo de este proyecto. Se detectó que la aplicación está subdividida en dos aplicaciones de Django, una destinada al perfil de estudiante que contiene las unidades y el avance, y otra destinada al profesor, es decir, el módulo editor.

Una vez detectado como estaba diseñada la aplicación se comenzó con la subsanación de errores detectados e implementación de pequeñas mejoras:

- Logout desde editor no redireccionaba a la página principal
- No funciona barra de administración desde estadísticas
- Unificar cabecera del módulo administrador con la de usuario
- Unificación de todos los colores de la página
- Error de la página al registrar un correo ya existente
- Los alumnos podían ver los cursos ocultos, pero no acceder a estos
- Controlar la inserción de unidades
- Introducción de la opción para abandonar cursos
- Centrar todas las ventanas emergentes de la página, los textos y los botones
- Cambio del botón cerrar sesión por un desplegable bajo el Nick de usuario

Después de todos estos cambios, y ya con un buen manejo de cómo funciona toda la aplicación se comenzó con el desarrollo en sí del objetivo principal de este trabajo.

Para comenzar se implementó la ampliación de la base de datos que ya hemos visto en el apartado anterior, todo sin modificar nada de las existentes. Con este cambio ya hecho, ya se podía comenzar a desarrollar la interfaz que va a hacer posible la creación de exámenes.

Para la interfaz, se comienza a insertar los botones necesarios en el panel de administración del curso. Para hacerlo de la manera más sencilla e intuitiva posible, la creación de exámenes es muy similar a la creación de una unidad, por lo que el botón de crear examen está al lado del botón crear unidad.

Con esto se crea la URL específica para la página de edición del examen, en esta se puede editar tanto el título como las preguntas que componen el examen. Para la creación de preguntas también se crea una URL nueva para su modificación y creación de las respuestas.

Una vez tenemos la interfaz para crear exámenes, y hemos comprobado que efectivamente se crean en la base de datos, es el momento de pasar a la implementación de la parte del usuario.

Al igual que para la creación de exámenes, la interfaz para el usuario se basa en cómo se muestran las unidades para mostrar los exámenes en la misma lista, de manera que el examen que hace referencia a una unidad se muestra en la lista después de esta.

Para la vista del examen se crea una URL donde se van a mostrar las 10 preguntas del examen que provienen del algoritmo de selección visto en el apartado de Diseño. Las preguntas son tipo test y exclusivas, por lo que solo se puede responder a una opción. De esta página no se puede salir salvo que se cierre el navegador, esto se ha programado así dado que una vez que se comienza el examen hay que acabarlo.

Cada URL que se ha mencionado tiene asociada una función Python de backend en donde se reciben los datos desde la vista, se opera según la funcionalidad que debe implementar y se redirecciona o renderiza la siguiente página. Por ejemplo, cuando se envía el examen a corregir, la función recibe todos los datos por medio de un formulario POST (HTTP), con estos opera y va comprobando si la respuesta dada es la correcta. Una vez acabado todo el cálculo renderiza otra vez el HTML del examen, pero esta vez para mostrar el resultado final de la corrección.

Una vez llegado a este punto, el desarrollo se da por terminado, y comienzan las pruebas para comprobar el correcto funcionamiento de todo el módulo de exámenes. Estas están detalladas en el siguiente apartado.

5 Integración, pruebas y resultados

Con el diseño y el desarrollo ya explicado y realizado nos disponemos a comprobar que todo funciona como se espera y hablar de la integración de este nuevo módulo en la plataforma original.

5.1 Integración

Al comienzo del desarrollo de este proyecto la plataforma ya estaba en funcionamiento, por lo que todo el diseño y desarrollo se hizo desde local para no interponerlo al correcto funcionamiento de la plataforma.

Ahora una vez que todo el proyecto está acabado y probado sería el momento de hacer la integración. Esta integración actualmente no es posible, porque durante el desarrollo de este trabajo la plataforma continuó actualizándose. Realizar una integración completa ahora es imposible, si bien podría ser factible en un futuro incorporar alguna de las mejoras desarrolladas en este trabajo.

5.2 Pruebas

Las pruebas realizadas no han sido a nivel tecnológico, ni a nivel de rendimiento, puesto que la plataforma base no ha sido tocada en cuanto a nivel arquitectónico, ni se espera un aumento de usuarios por la creación del nuevo módulo.

Se han realizado pruebas en los cálculos de probabilidades y que el algoritmo diseñado de selección de preguntas para la generación personalizada de los exámenes funciona correctamente.

5.2.1 Pruebas al cálculo de probabilidades de las categorías

Para esta prueba vamos a usar a un usuario nuevo creado del que no se dispone ningún conocimiento sobre sus categorías, por tanto, para todas se va a suponer el dominio por defecto que es 0,1. A continuación éste va a realizar un examen del que solo responde 3 preguntas:

- 2 preguntas de la categoría Variables, ambas de dificultad Básica.
- 1 pregunta de la categoría Punteros, de dificultad Media.

Las dos preguntas de Variables las acierta y la pregunta de Punteros la falla. Lo que pretendemos con esto es comprobar que el cálculo almacenado de dominio para estas dos categorías es correcto comparándolo con la fórmula vista en el **apartado 3.2**.

Podemos consultar la tabla **learnedtags** donde están almacenados estos dominios, y obtenemos lo siguiente (Figura X)

	user_id integer	description character varying(50)	p_domain double precision
1	4	Punteros	0.0270270270270271
2	4	Variables	0.155172413793103

Figura 4: Dominios obtenidos por la plataforma

Ahora vamos a comprobar que haciendo el cálculo manual obtenemos lo mismo. Sabiendo que las preguntas de Variables eran básicas y que no teníamos conocimiento previo del alumno. Primero calculamos el nuevo dominio gracias al primer acierto, y a continuación el dominio por el segundo acierto:

$$\begin{aligned} Z &= [P(\text{acierto}|\text{variables}) * P(\text{variables})] \\ &\quad + [P(\text{acierto}|\text{no variables}) * P(\text{no variables})] \\ &= [0.9 * 0.1] + [0.7 * 0.9] = 0.72 \end{aligned}$$

$$\begin{aligned} P(\text{variables actualizado}) \\ = [P(\text{acierto}|\text{variable}) * P(\text{variables})]/Z = [0.9 * 0.1]/0.72 = 0.125 \end{aligned}$$

Por tanto, después del primer acierto tenemos un dominio de 0.125 para la categoría de variables. Pero como tenemos otro acierto para la categoría, tenemos que hacer una vez más el mismo procedimiento, pero esta vez teniendo en cuenta que el dominio de la categoría ya no es 0.1 como antes:

$$\begin{aligned} Z &= [0.9 * 0.125] + [0.7 * (1 - 0.125)] = 0.725 \\ P(\text{variables actualizado}) &= 0.9 * 0.125/0.725 = 0.1551724 \end{aligned}$$

Este si es el dominio final después de la prueba para la categoría de variables, y podemos comprobar que efectivamente es el mismo obtenido en el nuevo módulo.

Ahora vamos a comprobar que con el fallo para una pregunta de categoría punteros de nivel medio también funciona correctamente. Igual que para la categoría anterior partimos sin conocer el dominio del alumno y, por tanto, se toma como dominio el valor por defecto de 0.1:

$$\begin{aligned} Z &= [1 - P(\text{acierto}|\text{punteros}) * P(\text{punteros})] \\ &\quad + [1 - P(\text{acierto}|\text{no punteros}) * P(\text{no punteros})] \\ &= [(1 - 0.9) * 0.1] + [(1 - 0.6) * 0.9] = 0.37 \end{aligned}$$

$$\begin{aligned} P(\text{punteros actualizado}) \\ = 1 - [(1 - P(\text{acierto}|\text{no punteros})) * P(\text{no punteros})]/Z \\ = 1 - [(1 - 0.6) * 0.9]/0.37 = 0.027027 \end{aligned}$$

Una vez más, vemos que para el fallo de la pregunta de punteros se ha calculado correctamente comparando con lo obtenido en nuestra plataforma (Figura X).

Con esto comprobamos que el algoritmo programado para la actualización de las probabilidades de acierto sigue la formula. Además, podemos comprobar que conforme se van realizando aciertos sobre la misma categoría el valor del dominio va creciendo, y con los fallos se va disminuyendo, siendo el valor neutro 0.1, que es el conocimiento por defecto.

5.2.2 Pruebas a la selección de preguntas

Para realizar esta prueba vamos a partir de un alumno del que no se dispone ningún conocimiento previo. Se matricula en un curso cuyo examen está compuesto por 45 preguntas divididas en 3 categorías.

Categoría	Nivel	Número
Punteros	Básico	8
	Medio	5
	Avanzado	2
Variables	Básico	8
	Medio	5
	Avanzado	2
Bucles	Básico	8
	Medio	5
	Avanzado	2

Tabla 4: Bolsa de preguntas del examen de prueba

El alumno se dispone a realizar el examen, y como no se tiene ningún conocimiento previo, la selección de 10 preguntas se hace de manera aleatoria como ya hemos visto en el **apartado 3.3**. Accedemos al examen y comprobamos en la terminal cual ha sido la selección realizada. Para poder ver las preguntas mediante el terminal se ha introducido un print en la función correspondiente del fichero **tutorial/views.py** con el propósito de facilitar las comprobaciones en esta prueba.

```
(55)ExamQuestion Categoría Punteros - Básica 8 TAG: B_32 Exam number: 1
(60)ExamQuestion Categoría Punteros - Media 5 TAG: M_32 Exam number: 1
(48)ExamQuestion Categoría Punteros - Básica 1 TAG: B_32 Exam number: 1
(63)ExamQuestion Categoría Variables - Básica 1 TAG: B_33 Exam number: 1
(59)ExamQuestion Categoría Punteros - Media 4 TAG: M_32 Exam number: 1
(72)ExamQuestion Categoría Variables - Media 2 TAG: M_33 Exam number: 1
(84)ExamQuestion Categoría Bucles - Básica 7 TAG: B_34 Exam number: 1
(71)ExamQuestion Categoría Variables - Media 1 TAG: M_33 Exam number: 1
(66)ExamQuestion Categoría Variables - Básica 4 TAG: B_33 Exam number: 1
(81)ExamQuestion Categoría Bucles - Básica 4 TAG: B_34 Exam number: 1
```

Figura 5: Prueba Examen Aleatorio

Podemos comprobar que se han seleccionado 4 preguntas de punteros, 4 preguntas de variables y 2 de bucles. Para comprobar que la selección se realiza conforme a los dominios vamos a completar este examen. Se fallan todas las preguntas de punteros, se aciertan todas las preguntas de variables, y de bucles se acierta y se falla una. Después de enviar el examen, procedemos a volver a realizarlo y ver que selección se nos genera.

```
(62)ExamQuestion Categoría Punteros - Avanzada 2 TAG: A_32 Exam number: 1
(56)ExamQuestion Categoría Punteros - Media 1 TAG: M_32 Exam number: 1
(51)ExamQuestion Categoría Punteros - Básica 4 TAG: B_32 Exam number: 1
(54)ExamQuestion Categoría Punteros - Básica 7 TAG: B_32 Exam number: 1
(61)ExamQuestion Categoría Punteros - Avanzada 1 TAG: A_32 Exam number: 1
(52)ExamQuestion Categoría Punteros - Básica 5 TAG: B_32 Exam number: 1
(49)ExamQuestion Categoría Punteros - Básica 2 TAG: B_32 Exam number: 1
(91)ExamQuestion Categoría Bucles - Avanzada 1 TAG: A_34 Exam number: 1
(89)ExamQuestion Categoría Bucles - Media 4 TAG: M_34 Exam number: 1
(58)ExamQuestion Categoría Punteros - Media 3 TAG: M_32 Exam number: 1
```

Figura 6: Prueba Examen con Conocimientos

Se observa que no se ha repetido ninguna pregunta con respecto al examen anterior, esto se comprueba comparando los números a la izquierda de cada línea mostrados entre paréntesis. Estos son los id únicos para cada pregunta.

Ahora vemos que en este segundo examen tenemos 8 preguntas de punteros, esto es debido a que es la única categoría que tenemos por debajo de 0.1 porque en el examen anterior hemos fallado todas. El algoritmo de selección reparte el 80% para las categorías inferiores a 0.1, y como es la única categoría que cumple esta condición tenemos las 8 preguntas de punteros. El 20% restante se reparte entre las otras categorías aleatoriamente. En este caso se han seleccionado 2 de bucles, pero podrían haber sido 2 de variables o 1 de bucles y 1 de variables.

Con esto hemos demostrado que el algoritmo funciona correctamente y que en conjunto con el cálculo de probabilidades para las categorías la selección se hace de manera inteligente y personalizada según las necesidades de cada estudiante.

6 Conclusiones y trabajo futuro

6.1 Conclusiones

En este documento se ha ido explicando y presentando todo el proceso de diseño y desarrollo del proyecto así como los problemas encontrados durante estos. Con todo lo visto en este documento podemos extraer una serie de conclusiones.

La plataforma continúa teniendo un futuro prometedor, puesto que con esta actualización se pone a la altura de otras plataformas existentes introduciendo la evaluación adaptada a las necesidades de los estudiantes. Las futuras ampliaciones de funcionalidad serán sencillas puesto que este desarrollo ha mantenido la modularidad con la que se diseñó la primera versión de la plataforma. Como extra, para facilitar las futuras ampliaciones, se ha comentado todo el código fuente de la plataforma con el objetivo de que todo esté bien documentado.

En cuanto a la tecnología empleada, con Django nos aseguramos de que el futuro de la plataforma es muy prometedor, puesto que este framework está a la vanguardia y en constante actualización.

Con los conocimientos adquiridos durante el grado de ingeniería informática es posible tomar las riendas de un proyecto ya existente y poder desarrollar sobre él.

Asignaturas	
EDAT	<ul style="list-style-type: none">• Diseño de Bases de Datos• Optimización de Bases de Datos• Implementación con PostgreSQL
ADSOF	<ul style="list-style-type: none">• Aprendizaje del paradigma de programación Orientación a Objetos
PADSOF	<ul style="list-style-type: none">• Introducción al Modelo Vista Controlador (MVC)• Implementación de una aplicación usando MVC
SI	<ul style="list-style-type: none">• Introducción a la programación web (HTML, CSS...)• Introducción a Servidores web
PSI	<ul style="list-style-type: none">• Aprendizaje Python• Introducción al framework Django

Tabla 5: Asignaturas imprescindibles para el desarrollo del proyecto

Es evidente que estas no son las únicas materias que durante el grado han hecho posible el desarrollo de este trabajo, todas y cada una de las asignaturas impartidas han influido de alguna manera para este trabajo.

Las asignaturas detalladas en la tabla son aquellas en las que se han tratado contenidos que han sido usados directamente en este trabajo. Pero sin duda alguna la asignatura que

más ha motivado el desarrollo de este proyecto es PSI, ya que fue la primera vez que se desarrolló usando un framework y publicó online la aplicación desarrollada.

6.2 Trabajo futuro

Con la realización de esta actualización de ENCODE para así obtener su segunda versión se han presentado nuevas ideas para futuras mejoras y ampliaciones. Sin embargo, actualmente esta segunda versión es muy competitiva y completa, pese a esto a continuación se van a plantear futuras mejoras que llevar a cabo en la plataforma.

- **Desarrollo de una aplicación Android**

Crear una aplicación Android que haga uso del servidor y la base de datos actual. Con esto se conseguiría un mayor impacto en los usuarios, pues estos podrían ponerse a realizar el curso en cualquier momento puesto que los dispositivos móviles siempre van con nosotros. También esta idea surge del gran aguje que están teniendo las aplicaciones móviles y el descenso del uso de los ordenadores.

- **Módulo de Usuario**

Añadir un nuevo módulo al igual que se ha añadido el explicado en este documento. Este nuevo módulo de usuario se encargaría de ampliar la información de perfil de cada usuario. En esta nueva sección el alumno podría tener un acceso a toda su información, avance en los cursos en los cuales se ha matriculado, ver las categorías en las que mejor se desenvuelve y en cuales necesita más refuerzo ...

- **Uso de un framework para el front-end**

Esta nueva versión de ENCODE ha mejorado su interfaz haciéndola más sencilla y visualmente más limpia, pero actualmente hay frameworks con los que realizar la interfaz. Por tanto, sustituir la actual interfaz desarrollada usando HTML, CSS y Semantic UI por un framework como **Angular** ó **React**.

Bibliografía

- [1] edX
URL: <https://www.edx.org/>
- [2] Udemy
URL: <https://www.udemy.com/>
- [3] CodeAcademy
URL: <https://www.edx.org/>
- [4] W3Schools
URL: <https://www.w3schools.com/>
- [5] StackOverflow
URL: <https://stackoverflow.com/>
- [6] Stack Exchange
URL: <https://stackexchange.com/>
- [7] Semantic UI
URL: <https://semantic-ui.com/>
- [8] JQuery
URL: <https://jquery.com/>
- [9] JQuery UI
URL: <https://jqueryui.com/>
- [10] Django Docs
URL: <https://docs.djangoproject.com/en/3.0/howto/deployment/>
- [11] Quora: Which database will work best with python Django?
URL: <https://www.quora.com/Which-database-will-work-best-with-python-Django>
- [12] Ace Editor
URL: <https://ace.c9.io/>

Glosario

Feedback	Devolución de una señal modificada a su emisor
Framework	Estructura conceptual y tecnológica de asistencia definida, con módulos concretos de software que puede servir tanto para el desarrollo como la organización del software.
back-end	Es la parte del desarrollo web que se encarga de que toda la lógica de una página web funcione correctamente. Se trata del conjunto de acciones que pasan por una web, pero no vemos.
front-end	Es la parte de una web que conecta e interactúa con los usuarios. Es la parte visible, la que muestra el diseño y el contenido.
Target	Objetivo, persona o cosa a la que se dirige una acción

Acrónimos

EDAT	Estructuras de Datos
ADSOF	Análisis y Diseño de Software
PAFSOF	Proyecto de Análisis y Diseño de Software
PSI	Proyecto de Sistemas Informáticos
SI	Sistemas Informáticos
MVC	Modelo Vista Controlador
CSS	Cascading Style Sheets (Hojas de estilos en cascada)
URL	Uniform Resource Locator (Localizador de Recursos Uniforme)
HTML	HyperText Markup Language (Lenguaje de marcas de hipertexto)

Anexos

A Manual de usuario

Este manual contempla ambos módulos de los que se compone la aplicación ENCODE. Se va a tratar de explicar de manera completa e intuitiva para que se haga un uso correcto de la herramienta, tanto desde el perfil de los estudiantes como de los profesores. En este manual solo se van a explicar las funcionalidades pertenecientes a esta nueva ampliación.

A.1 Módulo Tutorial

El módulo tutorial es el que está enfocado a los estudiantes, compuesto por diversas pantallas donde el estudiante se matricula en cursos, realiza sus unidades y los exámenes. Accedemos a <http://localhost:8000/> previamente habiendo lanzado la aplicación de Django.

En caso de disponer de una cuenta iniciamos sesión introduciendo nuestro correo electrónico y contraseña.

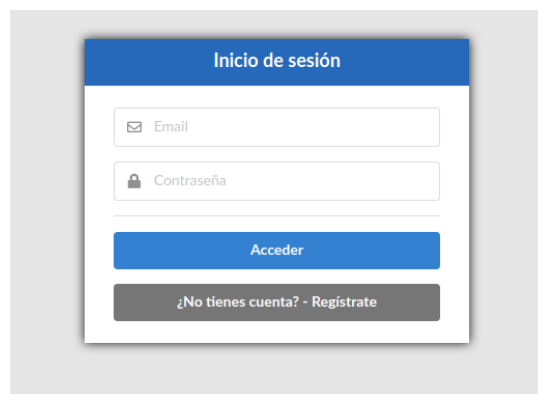
The image shows a web form titled 'Inicio de sesión' (Login) with a blue header. It contains two input fields: 'Email' with an envelope icon and 'Contraseña' (Password) with a lock icon. Below these fields is a blue button labeled 'Acceder' (Login) and a grey button labeled '¿No tienes cuenta? - Regístrate' (Don't you have an account? - Register).

Figura 7: Login ENCODE

En caso de no disponer aun de una cuenta es tan sencillo como pulsar el botón de ‘Registrarse’ y completar el formulario.

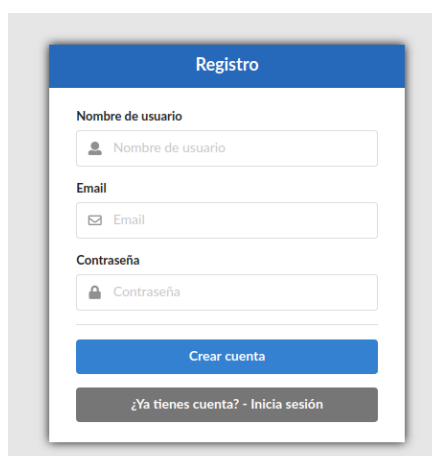
The image shows a web form titled 'Registro' (Registration) with a blue header. It contains three input fields: 'Nombre de usuario' (Username) with a person icon, 'Email' with an envelope icon, and 'Contraseña' (Password) with a lock icon. Below these fields is a blue button labeled 'Crear cuenta' (Create account) and a grey button labeled '¿Ya tienes cuenta? - Inicia sesión' (Do you already have an account? - Login).

Figura 8: Registro ENCODE

Una vez dentro observamos en la página principal dos listas, una con los cursos en los que estamos matriculados y otra con los cursos disponibles para matricularse.

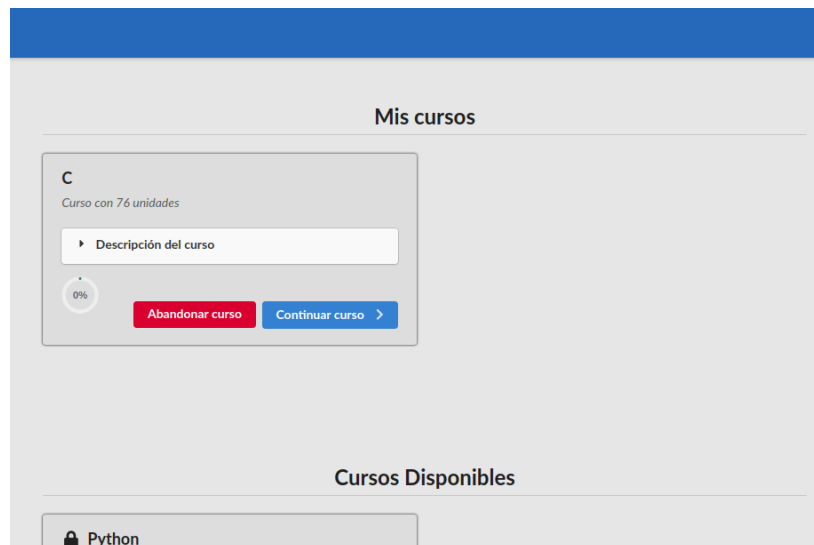


Figura 9: Página principal de ENCODE

Matricularse es tan sencillo como pulsar el botón verde de uno de los cursos disponibles.

En esta nueva versión es posible desmatricularse de un curso, pulsando el botón rojo “Abandonar curso” (**Figura 9**) y nos aparece una ventana emergente avisándonos si estamos seguros de desear abandonar, esta acción borra todo el avance que tuviéramos en el curso, y si queremos volver a matricularnos tendremos que comenzar el curso desde el principio (**Figura 10**)

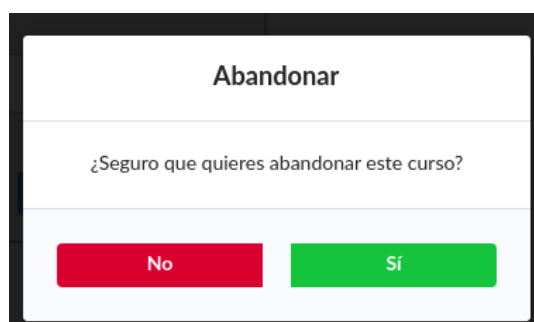


Figura 10: Confirmación abandono de curso

Volviendo al punto de querer continuar con un curso que ya tenemos matriculado, pulsamos sobre el botón azul que indica “Continuar curso”, y accedemos a la lista de unidades y exámenes que lo componen, en esta página podemos ver el avance que llevamos sobre el total del curso (**Figura 11**)

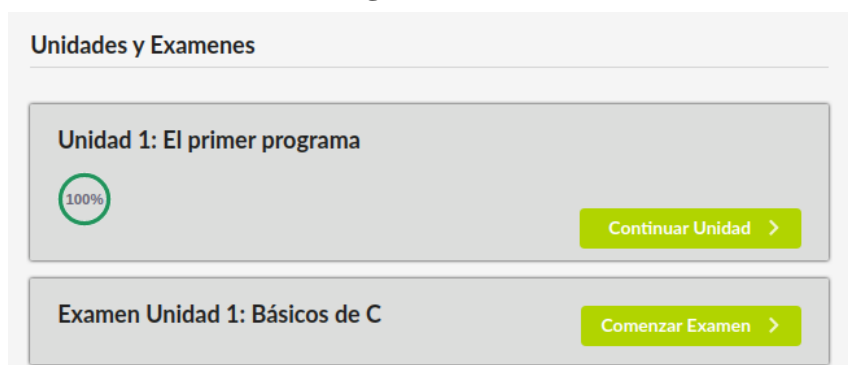


Figura 11: Lista de Unidades y Exámenes

Pulsando sobre el botón verde de “Continuar Unidad” o “Comenzar Unidad” (**Figura 11**) de cualquiera de los temas accedemos a los objetivos que componen la unidad y que debemos realizar. En este punto no se ha producido ningún cambio desde la versión anterior.

Desde esta pantalla con la lista de unidades y exámenes (**Figura 11**) podemos acceder a algunos de los exámenes que componen el curso pulsando “Comenzar examen” y accederemos a este.

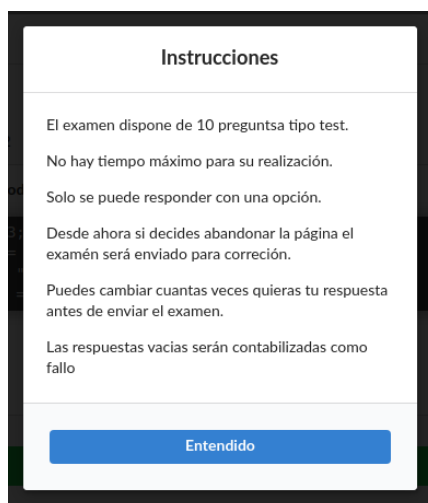


Figura 12: Instrucciones del examen

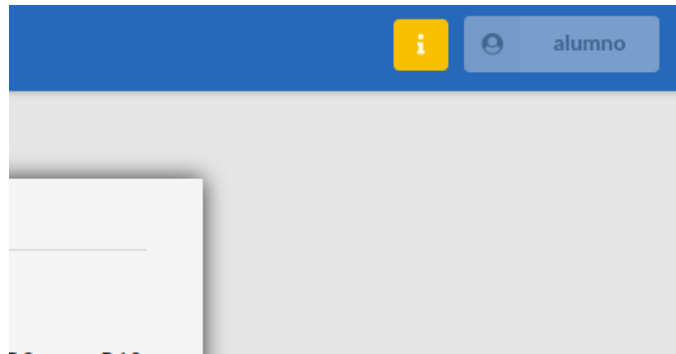


Figura 13: Botón instrucciones del examen

Al acceder al examen se muestra automáticamente un pop-up con las instrucciones para realizar el examen. Este mensaje puede ser mostrado en cualquier momento pulsando el botón amarillo situado en la parte superior derecha de la pantalla. (**Figura 13**)

Se debe ir respondiendo a las preguntas, que están separadas en pestañas. No es obligatorio contestar y solo se puede seleccionar una opción para cada pregunta. Una vez respondido todas las preguntas se pulsa el botón verde inferior “Enviar examen”.

Básicos de C

P.1 P.2 P.3 **P.4** P.5 P.6 P.7 P.8 P.9 P.10

¿Cuál es la salida por pantalla del siguiente programa?

```
int main (int argc, char *argv[]){
    if(3 + 2 == 5){
        printf("Suma Correcta");
        return 1;
    }
    return -1;
}
```

☐ Suma Correcta
☐ Error de compilación
☐ No produce ninguna salida

Enviar Examen

Figura 14: Formato Examen

Al momento se nos recargará la página mostrándonos las preguntas que se han respondido correctamente y las que han sido incorrectas. En un pop-up se nos indica cuantas preguntas hemos acertado y en el caso de algunos alumnos se muestra el feedback con las categorías que deben reforzar (**Figura 15**)

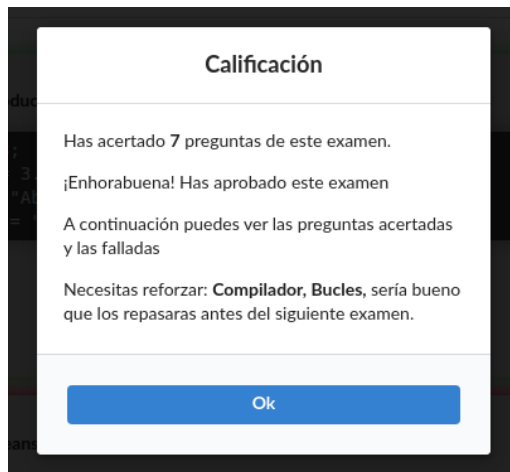


Figura 15: Feedback Examen

Al pulsar sobre el botón “Ok” cerramos esta ventana de feedback y puedes ver cuáles son las preguntas que has acertado, que aparecen remarcadas en verde, y las preguntas falladas que aparecen remarcadas en rojo (**Figura 16**). No se muestra cual es la solución correcta de las preguntas falladas para evitar que los alumnos se aprendan de memoria las respuestas, aunque es difícil que estas preguntas les vuelvan a aparecer otra vez en algún examen gracias al algoritmo de selección.

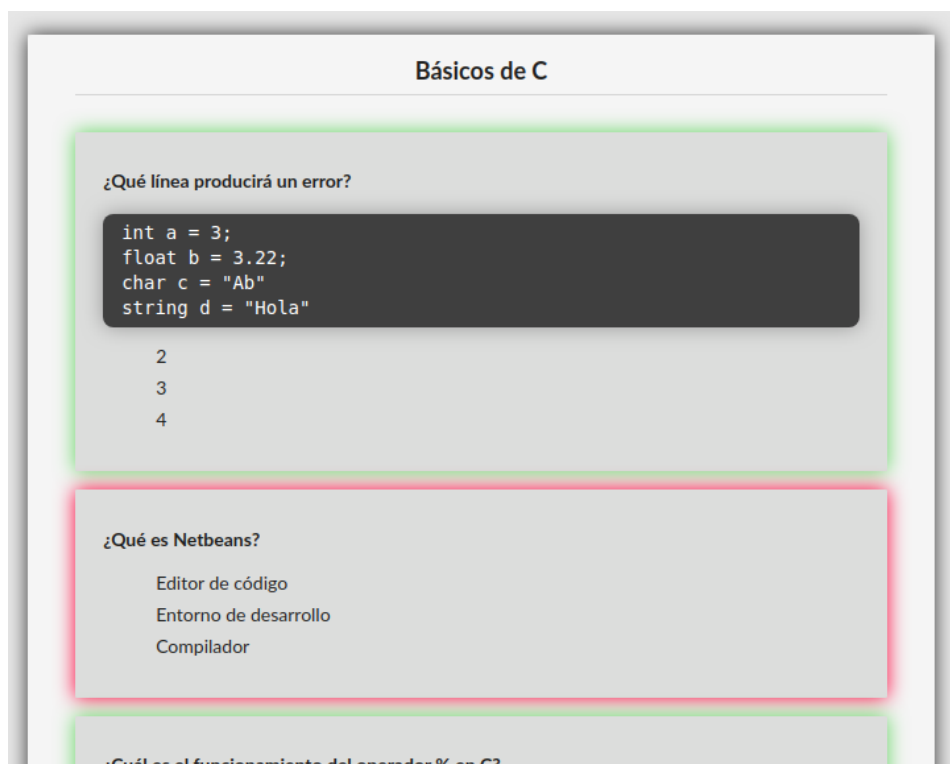


Figura 16: Revisión de un examen

Desde la corrección del examen para volver a la pantalla del curso pulsamos el botón “Volver al Curso” que está ubicado en la parte inferior en verde.

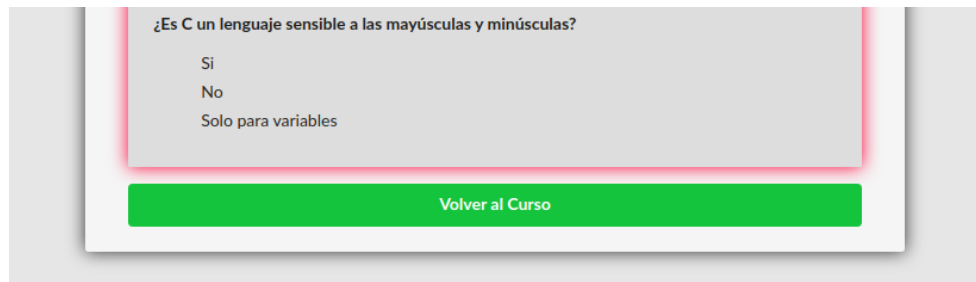


Figura 17: Volver al curso después de la corrección

Al volver a la pantalla del curso (**Figura 11**) volvemos a querer hacer el examen que acabamos de realizar, se nos va a mostrar un mensaje indicándonos que debemos esperar una hora antes de poder volver a realizarlo (**Figura 18**)

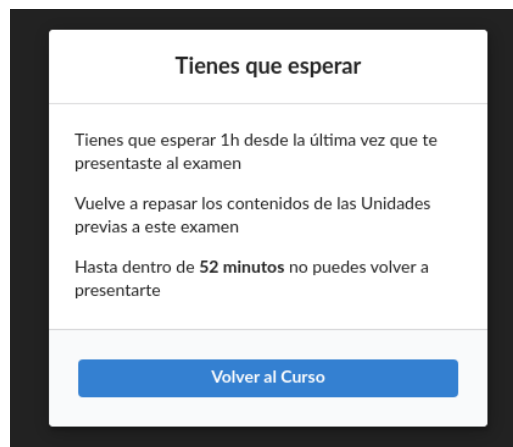


Figura 18: Mensaje reintento prematuro de examen

Una vez pasado el tiempo indicado podremos volver a realizar este examen, mientras tanto lo ideal es que el alumno realice los objetivos de la unidad otra vez, pero esta vez prestando atención en las categorías indicadas en la corrección. También puede seguir avanzando con el resto de las unidades y exámenes.

A.2 Módulo Editor

Para comenzar hay que estar logeado como un usuario administrador, en ese caso en la parte superior derecha, pulsando sobre nuestro nombre aparece un desplegable donde podemos ver la opción **Administración** (Figura 19) y accedemos a la página principal de edición (Figura X+1)

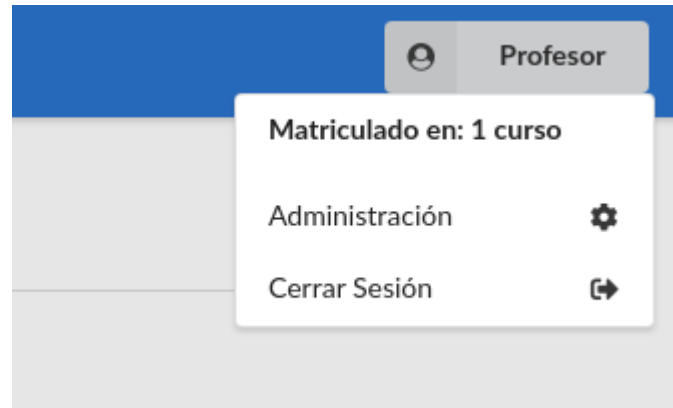


Figura 19: Menú desplegable para el usuario

En caso de intentar acceder a alguna de las páginas reservadas para los administradores se redirecciona a una pantalla indicándonos que el acceso está restringido (Figura 20)

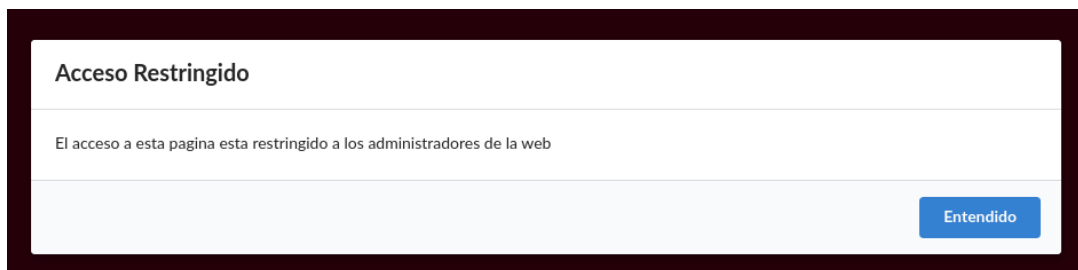


Figura 20: Mensaje de acceso restringido

Accedemos al curso sobre el que vamos a crear exámenes, en caso de no existir ninguno se crea uno nuevo.

The screenshot shows a web interface for editing course data. At the top, there is a navigation bar with 'Administración', 'Ir a objetivo', and 'Estadísticas'. Below this, the title 'Datos del Curso "C"' is displayed. The form contains two input fields: 'Título de curso' with the value 'C' and 'Email Profesor' with the value 'alejandro.sierra@uam.es'. A 'Descripción' field contains the text: 'Curso para aprender el lengua de programación C. Desarrollado en el año 1972 por Dennis Ritchie para UNIX un sistema operativo multiplataforma. El lenguaje C es del tipo lenguaje estructurado como son Pascal,'. Below the description are two buttons: 'Guardar cambios' and 'Editar Categorías'. At the bottom, there is a section titled 'Lista de unidades' with buttons 'Volver' and 'Nueva Unidad'. A single unit is listed: 'Unidad 1: El primer programa'.

Figura 21: Ventana para edición de un curso

Desde aquí podemos crear, editar y eliminar preguntas, el funcionamiento es muy similar al de unidades y exámenes. Como se ve en la **Figura 21** en la parte superior tenemos el cuadro de control para crear nuevas preguntas y en la parte inferior la lista de preguntas ya existentes dentro del examen que se están editando.

Además, desde esta página podemos acceder a las categorías que existen en el curso y poder editarlas. Pulsado el botón “Editar Categorías” nos lleva a la página donde podemos ver la lista de categorías e incluir todas aquellas que se desee. En la **Figura 22** vemos la lista de categorías existentes, pulsando sobre el botón “Nueva Categoría”, ubicado en la parte inferior derecha aparecerá un nuevo elemento en la lista vacío, una vez escrito el nombre de esta se pulsa el botón de “Guardar Categorías” ubicado en la parte inferior derecha para guardar todos los cambios realizados.

The screenshot shows a table titled 'Categorías del Curso C'. The table has two main columns: 'Descripción' and 'Acciones'. The 'Descripción' column lists five categories: 'Punteros', 'Memoria', 'Bucles', 'Variables', and 'Declaraciones'. The 'Acciones' column contains two buttons for each category: 'Editar' (orange) and 'Eliminar' (red). At the bottom of the table, there are two buttons: 'Nueva Categorías' (teal) and 'Guardar Categorías' (green).

Descripción	Acciones
Punteros	Editar Eliminar
Memoria	Editar Eliminar
Bucles	Editar Eliminar
Variables	Editar Eliminar
Declaraciones	Editar Eliminar

Nueva Categorías Guardar Categorías

Figura 22: Ventana para edición, creación y eliminación de categorías

Pulsando el botón rojo “Eliminar” de alguna de las categorías, se eliminará la categoría solo en el caso de que esta no esté siendo usada por alguna pregunta del curso. En caso de estar siendo usada saldrá un error como el mostrado en la (**Figura X**)

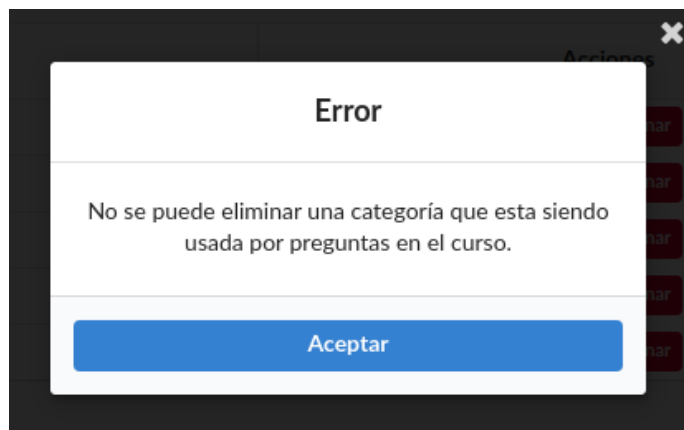


Figura 23: Error al eliminar una categoría

Una vez dentro de la edición del curso, podemos observar dos listas, la primera para las unidades, y la siguiente para los exámenes. Al lado del título “Lista de exámenes encontramos un botón “Nuevo Examen”. La plataforma limita a un examen por cada unidad, por tanto, cada examen esta enlazado a una unidad, y el máximo de exámenes es el número total de unidades que componen el curso. Una vez pulsado el botón de” Nuevo Examen”, nos aparece el desplegable (**Figura 24**), en este introducimos el título del examen, y seleccionamos de la lista la unidad que examina, en este desplegable solo aparecen las unidades que aún no disponen de un examen.

A screenshot of a form titled "Nuevo Examen". The form has a white background with a dark border. It contains two input fields: "Unidad del Examen" with a dropdown menu showing "1", and "Título del Examen" with a text input field containing "Básicos de C". At the bottom, there are two buttons: a red "Cancelar" button and a green "Crear" button.

Figura 24: Creación Examen

Una vez pulsamos crear, aparece el examen en la lista inferior de exámenes. El examen podemos publicarlo/ocultarlo, editarlo, y eliminarlo, al igual que se puede hacer con las unidades (**Figura 25**).

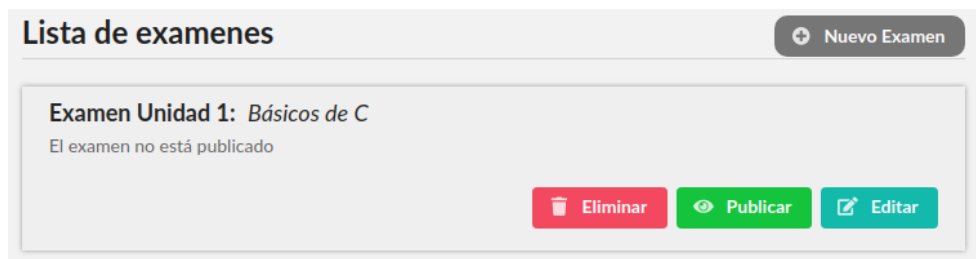


Figura 25: Examen recién creado

Si pulsamos sobre el botón editar nos redirige a la página desde la que vamos a poder configurar y crear las preguntas del examen.

Para crear una nueva pregunta completamos el campo enunciado, en el editor de código podemos introducir opcionalmente una porción de código complementaria al enunciado de la pregunta, y los tres campos de soluciones posibles y marcamos al menos una de ellas como correcta. A cada pregunta hay que asociarle un tipo de nivel en función de su complejidad (Básica, Media, Avanzada) además hay que seleccionar a que categoría pertenece. Estas dos últimas selecciones son utilizadas para el cálculo de probabilidades y el buen funcionamiento del algoritmo de selección.

Figura 26: Creación Pregunta

Con todos los campos completados se puede pulsar el botón “Crear”, si algún campo estuviera vacío o mal completado saltará un mensaje de error para que pueda ser completado correctamente.

Una vez creada la pregunta podremos verla en la lista inferior (**Figura 27**). Para el correcto funcionamiento de los exámenes lo recomendable es introducir unas 20 o más preguntas de categorías variadas que pertenezcan al contenido visto en la unidad asociada. De esta manera se asegura un buen repositorio de preguntas para que no se repitan. El mínimo establecido para poder publicar el examen es de 10 preguntas.



Preguntas

¿Cuál es la salida por pantalla del siguiente programa?

```
int main (int argc, char *argv[]){
    if(3 + 2 == 5){
        printf("Suma Correcta");
        return 1;
    }
    return -1;
}
```

- Suma Correcta
- Error de compilación
- No produce ninguna salida

[Eliminar](#) [Editar](#)

Figura 27: Pregunta recién creada

Las preguntas pueden ser editadas, para corregir pequeños errores, o hacer cambios para mejorar su comprensión. Pulsando el botón “Editar” de la pregunta deseada vamos a la página donde podemos editarla. Podemos editar el enunciado, el código si es que lo posee, y el valor de las tres opciones. No se permite cambiar cual es la correcta, puesto que esa pregunta puede haber sido respondida ya por usuarios. En el campo de cada posible respuesta aparece si es la solución o no. Cuando se hayan realizado los cambios deseados, se pulsa el botón “Guardar cambios” que está situado en la parte inferior en color verde. Si al guardar los cambios ocurre algún error, como puede ser que se haya dejado algún campo en blanco, aparecerá en pantalla el error indicando como solucionarlo. Si todo está bien, al guardar nos lleva a la página donde aparecen todas las preguntas pertenecientes al examen.

Administración

Ir a objetivo ▾

Estadísticas

Examen unidad 1: Básicos de C

Enunciado

¿Cuál es la salida por pantalla del siguiente programa?

Código

```
1 int main (int argc, char *argv[]){
2     if(3 + 2 == 5){
3         printf("Suma Correcta");
4         return 1;
5     }
6     return -1;
7 }
```

Respuesta correcta

Suma Correcta

Respuesta incorrecta

Error de compilación

Respuesta incorrecta

No produce ninguna salida

Guardar cambios

Figura 28: Edición de una pregunta

